Chương 4

Ngôn ngữ SQL

Nội dung trình bày

- Giới thiệu
- Định nghĩa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

Giới thiệu

- SQL (Structured Query Language)
  - Ngôn ngữ cấp cao
  - Được phát triển bởi IBM (1970s)
  - Được gọi là SEQUEL
  - Được ANSI công nhận và phát triển thành chuẩn
    - SQL-86
    - SQL-92
    - SQL-99
- SQL sử dụng thuật ngữ
  - Bảng ~ Quan hệ
  - Cột ~ Thuộc tính
  - Dòng ~ Bộ

Định nghĩa dữ liệu

- Là ngôn ngữ mô tả
  - Lược đồ cho mỗi quan hệ
  - Miền giá trị tương ứng của từng thuộc tính
  - Ràng buộc toàn vẹn
  - Chỉ mục trên mỗi quan hệ
- Gồm
  - CREATE / DROP DATABASE (tạo / xóa cơ sở dữ liệu)
  - CREATE / DROP / ALTER TABLE (tạo / xóa / sửa bảng)
  - CREATE / DROP DOMAIN (tạo / xóa miền giá trị)
  - …

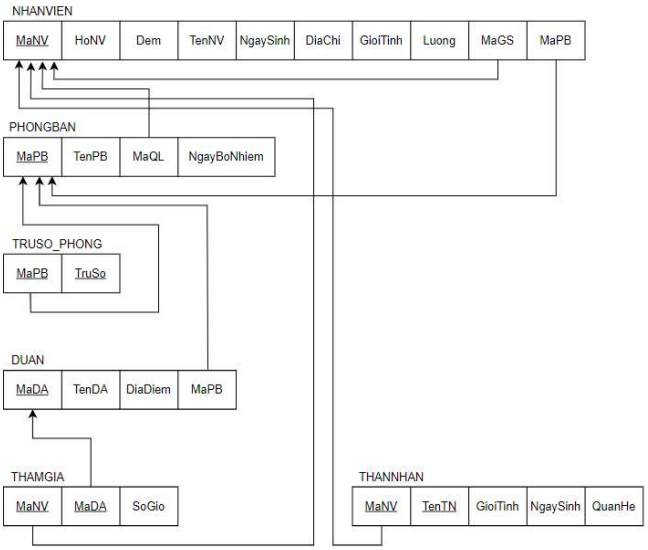
Kiểu dữ liệu (1)

- Số (numeric)
  - INTEGER, INT
  - SMALL INTEGER, SMALLINT
  - NUMERIC(p,s), NUMERIC(p)
  - DECIMAL(p,s), DECIMAL(p)
  - FLOAT(n), REAL
  - DOUBLE PRECISION
- Chuỗi ký tự (character string)
  - CHARACTER(n), CHAR(n)
  - CHARACTER VARYING(n), VARCHAR(n)

Kiểu dữ liệu (2)

- Chuỗi bit (bit string)
  - BIT(n)
  - BIT VARYING(n)
- Ngày giờ (datetime)
  - DATE gồm ngày, tháng và năm
  - TIME gồm giờ, phút và giây
  - TIMESTAMP gồm ngày và giờ

Lược đồ CSDL "Quản lý Công ty"



Tạo và xóa CSDL

- CREATE DATABASE <tên csdl>
  - create database CsdlCongty;
- DROP DATABASE <tên csdl> [RESTRICT | CASCADE]
  - drop database CsdlCongty;

Tạo bảng (1)

- CREATE TABLE <tên bảng> (
  <tên thuộc tính> <kiểu dữ liệu> [RBTV],
  <tên thuộc tính> <kiểu dữ liệu> [RBTV],
  ...
  [RBTV]
)
- Các loại RBTV
  - NOT NULL : không cho phép nhận giá trị null
  - NULL : cho phép nhận giá trị null
  - UNIQUE : yêu cầu các giá trị trong một cột phải khác nhau
  - DEFAULT : thiết lập giá trị mặc định
  - PRIMARY KEY : xác định khóa chính cho bảng
  - FOREIGN KEY / REFERENCES : xác định khóa ngoại cho bảng
  - CHECK : giới hạn phạm vi giá trị trong một cột

Tạo bảng (2)

- Ví dụ

```
create table PHONGBAN (
  MaPB          int primary key,
  TenPB         varchar(15) not null,
  MaQL          char(9),
  NgayBoNhiem   date
);
```

Tạo bảng (3)

- Đặt tên cho RBTV
  - CONSTRAINT <tên RBTV> <RBTV>
- Ví dụ

```
create table PHONGBAN (
   MaPB              int,
   TenPB             varchar(15) not null,
   MaQL              char(9),
   NgayBoNhiem       date,
   constraint PHONGBAN_PK primary key (MaPB),
);
```

Xóa bảng

- DROP TABLE <tên bảng> [RESTRICT | CASCADE]
- Ví dụ

```
drop table PHONGBAN;
```

Sửa bảng (1)

- ALTER TABLE <tên bảng> ADD COLUMN <tên thuộc tính> <kiểu dữ liệu> [<RBTV>]
  - Ví dụ
    - alter table NHANVIEN add column DThoai char(10);
- ALTER TABLE <tên bảng> DROP COLUMN <tên thuộc tính>
  - Ví dụ
    - alter table NHANVIEN drop column DThoai;
- ALTER TABLE <tên bảng> ALTER COLUMN <tên thuộc tính> <kiểu dữ liệu mới>
  - Ví dụ
    - alter table NHANVIEN alter column GioiTinh int;

Sửa bảng (2)

- ALTER TABLE <tên bảng> ADD CONSTRAINT <tên RBTV> <RBTV>
  - Ví dụ
    - alter table NHANVIEN add constraint NV_Luong check (Luong > 0);
- ALTER TABLE <tên bảng> DROP CONSTRAINT <tên RBTV>
  - Ví dụ
    - alter table NHANVIEN drop constraint NV_Luong;

Tạo và xóa miền giá trị

- CREATE DOMAIN <tên kiểu dữ liệu mới> AS <kiểu dữ liệu cơ bản>
  - Ví dụ
    - create domain DIACHI as varchar(100);
- DROP DOMAIN <tên kiểu dữ liệu>

Cập nhật dữ liệu (1)

- INSERT - Thêm dòng
  - INSERT INTO <tên bảng> [<danh sách thuộc tính>] VALUES <danh sách giá trị>
  - Ví dụ
    - insert into DUAN (MaDA, TenDA, DiaDiem, MaPB) values (1,'San pham X', 'Tan Binh',5);
    - insert into DUAN values (1,'San pham X', 'Tan Binh',5);

Cập nhật dữ liệu (2)

- DELETE - Xóa dòng
  - DELETE FROM <tên bảng> [WHERE <điều kiện>]
  - Ví dụ
    - delete from NHANVIEN where MaPB = 5;
    - delete from NHANVIEN;

Cập nhật dữ liệu (3)

- UPDATE - Sửa dữ liệu
  - UPDATE <tên bảng> SET
    <tên thuộc tính>=<giá trị mới>,
    <tên thuộc tính>=<giá trị mới>,
    ...
    [WHERE <điều kiện>]
  - Ví dụ
    - update DUAN set DiaDiem='Hanoi', MaPB=4 where MaDA=10;
    - update NHANVIEN set Luong=1.1*Luong;

Truy vấn dữ liệu

- Cú pháp
  - SELECT <dsc>
    FROM <dsb>
    [WHERE <đk>]
    [GROUP BY <dsc nhóm>]
    [HAVING <đk nhóm>]
    [ORDER BY <dsc sắp xếp>]
- Kết quả của lệnh truy vấn là một bảng.
- Bảng kết quả của lệnh SELECT cho phép có nhiều dòng trùng nhau.

- Trong đó
  - <dsc>: tên các cột của bảng kết quả.
  - <dsb>: tên các bảng liên quan đến câu truy vấn.
  - <đk>: biểu thức logic để chọn hoặc ghép các dòng.
  - <dsc nhóm>: tên các cột để gom nhóm các dòng.
  - <đk nhóm>: biểu thức logic để chọn nhóm các dòng.
  - <dsc sắp xếp>: tên các cột và thứ tự sắp xếp tương ứng để sắp xếp các dòng của bảng kết quả.

Trích các dòng của một bảng (1)

- select * from R where <đkc>
- Để trích các dòng của bảng R thỏa điều kiện chọn.

R	A	B	C	D
	α	α	1	7
	α	β	5	7
	β	β	12	3
	β	β	23	10

A=B AND D>5 →

A	B	C	D
α	α	1	7
β	β	23	10

Trích các dòng của một bảng (2)

- Điều kiện chọn
  - Chứa các mệnh đề logic có dạng
    - <tên cột> <phép toán so sánh> <hằng số>.
    - <tên cột> <phép toán so sánh> <tên cột>.
  - Phép toán so sánh: =, <, ≤, >, ≥, ≠.
  - Các mệnh đề logic được kết nối bởi: AND, OR, NOT.
- Tìm các nhân viên làm việc trong phòng số 4.

```
select * from NHANVIEN where MaPB = 4;
```

Trích các cột của một bảng

- select <dsc> from R
- Để trích các cột của bảng R.

R	A	B	C	D
	α	α	1	7
	α	β	5	7
	β	β	12	3
	β	β	23	10

A,D →

A	D
α	7
β	3
β	10

- Cho biết họ tên, giới tính và mức lương của các nhân viên.

```
select HoNV, Dem, TenNV, GioiTinh, Luong
from NHANVIEN;
```

Kết nối các bảng (1)

- Kết nối không điều kiện :
  - select * from R, S
- Để kết nối các dòng của bảng R với các dòng của bảng S.

R	A	B	C
	α	α	1
	α	β	5
	β	β	12

S	D	E
	1	7
	5	7

→

A	B	C	D	E
α	α	1	1	7
α	α	1	5	7
α	β	5	1	7
α	β	5	5	7
β	β	12	1	7
β	β	12	5	7

Kết nối các bảng (2)

- Kết nối có điều kiện :
  - select * from R, S where <đkk>
- Để kết nối một dòng của bảng R với một dòng của bảng S thỏa điều kiện kết .

R	A	B	C
	α	α	1
	α	β	5
	β	α	5
	β	β	12

S	E	F
	α	1
	α	4
	β	12

A=E AND C<F →

A	B	C	E	F
α	α	1	α	4
β	α	5	β	12

Kết nối các bảng (3)

- Điều kiện kết
  - Chứa các mệnh đề logic có dạng
    - <tên cột R> <phép toán so sánh> <tên cột S>.
  - Phép toán so sánh: =, <, ≤, >, ≥, ≠.
  - Các mệnh đề logic được kết nối bởi: AND.
- Cho biết tên các trưởng phòng.

```
select HoNV, Dem, TenNV
from NHANVIEN, PHONGBAN
where MaQL = MaNV;
```

UNION, EXCEPT, INTERSECT (1)

- SQL có cài đặt các phép toán tập hợp cho 2 bảng kết quả
  - UNION (Hội)
  - INTERSECT (Giao)
  - EXCEPT (Hiệu)
- 2 bảng tham gia phép toán phải có tính khả hợp
- Bảng kết quả của phép toán là một tập hợp, nghĩa là các dòng trùng nhau sẽ bị loại.
- Để giữ lại các dòng trùng nhau thì kết hợp phép toán với ALL
  - UNION ALL
  - EXCEPT ALL
  - INTERSECT ALL

UNION, EXCEPT, INTERSECT (2)

- (SELECT <dsc> FROM <dsb> WHERE <đk>)
  UNION [ALL]
  (SELECT <dsc> FROM <dsb> WHERE <đk>)
- (SELECT <dsc> FROM <dsb> WHERE <đk>)
  EXCEPT [ALL]
  (SELECT <dsc> FROM <dsb> WHERE <đk>)
- (SELECT <dsc> FROM <dsb> WHERE <đk>)
  INTERSECT [ALL]
  (SELECT <dsc> FROM <dsb> WHERE <đk>)

UNION, EXCEPT, INTERSECT (3)

- Truy xuất các dự án do phòng số 5 hoặc phòng số 1 điều phối

```
(select * from DUAN where MaPB = 5)
union
(select * from DUAN where MaPB = 1);
```

- Cho biết mã số các nhân viên không tham gia dự án nào

```
(select MaNV from NHANVIEN)
except
(select MaNV from THAMGIA);
```

Đặt tên trong SELECT và FROM

- Đặt tên trong mệnh đề SELECT:

```
select sum(Luong) as 'Tong luong cac nhan vien'
from NHANVIEN;
```

- Đặt tên trong mệnh đề FROM:

```
select NV.Ho, NV.Ten. PB.Ten
from NHANVIEN as NV, PHONGBAN
    as PB where NV.MaPB = PB.MaPB;
```

Phép toán số học trong câu truy vấn

- +, –, *, / có thể áp dụng cho các *giá trị số* trong mệnh đề SELECT

```
select 1.1 * Luong as 'Luong moi' from NHANVIEN
```

NHANVIEN	...	Luong	...
		40000	
		30000	
		38000	
		25000	

Luong moi
44000
33000
41800
27500

- +, - có thể áp dụng cho các *giá trị kiểu ngày giờ*

```
select NgayBoNhiem + 150 as 'Cong Ngay' from
PHONGBAN;
```

Phép toán so sánh và logic trong câu truy vấn

- Dùng để xây dựng các biểu thức điều kiện.
  - =, <, ≤, >, ≥, <>
  - BETWEEN <giá trị> AND <giá trị>
  - AND, OR, NOT
- Tìm các nhân viên phòng số 5 có lương giữa 30.000 và 40.000
  - select * from NHANVIEN
    where (Luong >= 30000) and (Luong <= 40000) and
    MaPB = 5;
  - select * from NHANVIEN
    where (Luong between 30000 and 40000) and
    MaPB = 5;

So sánh chuỗi (1)

- LIKE <mẫu đối sánh> [ESCAPE <kí tự thoát>]
- Mẫu đối sánh là một chuỗi ký tự để so sánh, có thể chứa các kí tự đặc biệt
  - % - thay cho *một đoạn* ký tự tùy ý.
  - _ - thay cho *một* ký tự tùy ý.
- Ký tự thoát là ký tự để loại bỏ chức năng của ký tự đặc biệt % và _.
  - Có thể dùng ký tự bất kỳ miễn nó không có mặt trong mẫu đối sánh.

So sánh chuỗi (2)

- Tìm các nhân viên họ Nguyen

```
select *
from NHANVIEN
where HoNV LIKE 'Nguyen%';
```

- Tìm các nhân viên họ Nguyen_

```
select *
from NHANVIEN
where HoNV LIKE 'Nguyen\_% escape '\';
```

Tính toán và so sánh với NULL

- Tính toán và so sánh với NULL
  - null + 3 $\rightarrow$ null.
  - null > 3 $\rightarrow$ unknown.
- SQL cung cấp 2 phép toán để kiểm tra một giá trị có là NULL hay không
  - IS NULL
  - IS NOT NULL
- Tìm các nhân viên không có người giám sát
  - select * from NHANVIEN where MaGS is null;

Logic 3 chân trị

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

NOT	
TRUE	FALSE
FALSE	TRUE
UNKNOWN	UNKNOWN

Loại bỏ các dòng trùng nhau

- SQL không loại bỏ các dòng trùng nhau trong bảng kết quả vì
  - Tốn thời gian so sánh và sắp xếp.
  - Có thể cần cho các truy vấn thống kê dữ liệu.
- Sử dụng DISTINCT để loại bỏ các dòng trùng nhau trong kết quả
- Liệt kê các mức lương của các nhân viên

  select distinct Luong from NHANVIEN;

Hàm SUM, MAX, MIN, AVG, COUNT

- SQL cài đặt 5 hàm để thực hiện các tính toán thống kê dữ liệi trên tập hợp các dòng:
  - SUM(<tên cột>) - tính tổng các giá trị trong cột
  - MAX(<tên cột>) - lấy giá trị lớn nhất trong cột
  - MIN(<tên cột>) - lấy giá trị nhỏ nhất trong cột
  - AVG(<tên cột>) - lấy giá trị trung bình trong cột
  - COUNT(*) - đếm số dòng của bảng
  - COUNT(<tên cột>) - đếm số giá trị khác null trong cột
- Chỉ được xuất hiện trong mệnh đề SELECT hoặc HAVING
- Ví dụ

```
select sum(Luong) AS TongLuong,
       max(Luong) AS LuongCaonhat,
       min(Luong) AS LuongThapnhat,
       avg(Luong) AS LuongTB
from NHANVIEN
```

Gom nhóm các dòng (1)

- Mệnh đề GROUP BY dùng để gom nhóm các dòng trong bảng kết quả.
- Cú pháp cơ bản

```
SELECT <dsc nhóm> [, <dshth>]
FROM <dsb>
[WHERE <đk>]
GROUP BY <dsc nhóm>
[HAVING <đk nhóm>]
```

- Trong đó
  - <dsc nhóm>: tên các cột để gom nhóm
  - <dshth>: các hàm tập hợp để thống kê dữ liệu trên các nhóm
  - <đk nhóm>: điều kiện để trích các nhóm

Gom nhóm các dòng (2)

- Mệnh đề GROUP BY
  - Các cột trong mệnh đề SELECT (trừ những cột trong các hàm tập hợp) phải xuất hiện trong mệnh đề GROUP BY
- Mệnh đề HAVING
  - Sử dụng các hàm tập hợp trong mệnh đề SELECT trong điều kiện nhóm.
  - Điều kiện nhóm dùng để chọn lọc trên nhóm, không dùng để chọn lọc trên từng dòng.
  - Sau khi gom nhóm điều kiện trên nhóm mới được ước lượng.

Gom nhóm các dòng (3)

- Thứ tự thực hiện lệnh truy vấn có mệnh đề GROUP BY và HAVING
  1. Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
  2. Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
  3. Áp dụng các hàm tập hợp (nếu có) trong mệnh đề SELECT cho mỗi nhóm
  4. Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
  5. Trích các giá trị của các cột và hàm tập hợp trong mệnh đề SELECT đại diện cho từng nhóm thỏa điều kiện trong mệnh đề HAVING

Gom nhóm các dòng (3)

- Với mỗi phòng, cho biết số dự án phòng đó điều phối.

```
select MaPB, count(MaDA) as 'So du an'
from DUAN
group by MaPB;
```

DUAN	TenDA	MaDA	Diadiem	Phong
	Sản phẩm X	1	Tân Bình	5
	Sản phẩm Y	2	Thủ Đức	5
	Sản phẩm Z	3	Phú Nhuận	5
	Tin học hóa	10	Gò Vấp	4
	Phúc lợi	30	Gò Vấp	4
	Tái tổ chức	20	Phú Nhuận	1

Phong	So du an
5	3
4	2
1	1

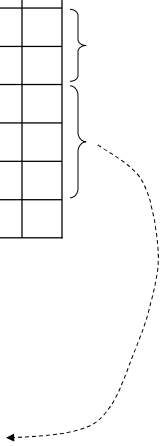
Gom nhóm các dòng (4)

- Cho biết mã số, tên dự án và số nhân viên tham gia đối với những dự án có nhiều hơn 2 nhân viên tham gia.

```
select DA.MaDA, DA.TenDA, count(*) as 'So nhan vien
tham gia'
from DUAN as DA, THAMGIA as TG
where DA.MaDA=TG.MaDA
group by DA.MaDA, DA.TenDA
having count(*) > 2;
```

Gom nhóm các dòng (5)

DA.TenDA	DA.MaDA	...	TG.MaNV	TG.MaDA	...
Sản phẩm X	1		123456789	1	
Sản phẩm X	1		453453453	1	
Sản phẩm Y	2		123456789	2	
Sản phẩm Y	2		453453453	2	
Sản phẩm Y	2		333445555	2	
...					



DA.MaDA	DA.TenDA	So nhan vien
2	Sản phẩm Y	3

Sắp xếp kết quả (1)

- Mệnh đề ORDER BY dùng để sắp xếp kết quả truy vấn
- Cú pháp cơ bản
  - SELECT <dsc>
    FROM <dsb>
    [WHERE <đk>]
    ORDER BY <dsc sắp xếp>
  - <dsc sắp xếp>: danh sách các cặp (tên cột, thứ tự sắp xếp).
  - Thứ tự: ASC (tăng dần), DESC (giảm dần), mặc định là ASC.

Sắp xếp kết quả (2)

- Với mỗi nhân viên, cho biết mã nhân viên và mã dự án mà nhân viên đó tham gia. Sắp xếp kết quả theo thứ tự tăng dần của mã nhân viên và giảm dần của mã dự án.

```
select MaNV, MaDA
from THAMGIA
order by MaNV asc, MaDA desc
```

MaNV	MaDA
123456789	2
123456789	1
333445555	20
333445555	10
333445555	3
333445555	2
...	

Truy vấn lồng (1)

- Một lệnh truy vấn gọi một lệnh truy vấn khác trong biểu thức điều kiện chọn.

Lệnh truy vấn cha

```
SELECT <dsc>
FROM <dsb>
WHERE <so sánh tập hợp> (
    SELECT <dsc>
    FROM <dsb>
    WHERE <đk>)
```

Lệnh truy vấn con

Truy vấn lồng (2)

- Các lệnh SELECT có thể lồng nhau ở nhiều mức.
- Lệnh truy vấn con thường trả về một tập các giá trị vô hướng.
- Lệnh truy vấn con chỉ xuất hiện trong mệnh đề WHERE của lệnh truy vấn cha.
- So sánh tập hợp thường sử dụng các phép toán IN, NOT IN, ALL, ANY, EXISTS, NOT EXISTS.

IN, ALL, ANY, EXISTS (1)

- Phép toán
  - IN - kiểm tra sự tồn tại của một giá trị trong một tập hợp.
  - ALL - so sánh một giá trị với tất cả các giá trị của tập hợp.
  - ANY - so sánh một giá trị với một giá trị nào đó của tập hợp.
  - ALL, ANY được kết hợp với các phép toán so sánh =, <, ≤, >, ≥, <>.
  - EXISTS - kiểm tra sự tồn tại của kết quả của một lệnh truy vấn.
- Cú pháp
  - <tên cột> IN (<lệnh truy vấn>)
  - <tên cột> <phép toán so sánh> ALL (<lệnh truy vấn>)
  - <tên cột> <phép toán so sánh> ANY (<lệnh truy vấn>)
  - EXISTS (<lệnh truy vấn>)

IN, ALL, ANY, EXISTS (2)

- Tìm các nhân viên của phòng số 2 và 5.

```
select *
from NHANVIEN
where MaPB in (2, 5);
```

Tập hợp tường minh

- Tìm các nhân viên của phòng ‘Nghiên cứu’.

```
select *
from NHANVIEN
where MaPB = (select MaPB from PHONGBAN
              where TenPB = ‘Nghiên cứu’);
```

Lệnh truy vấn vô hướng

IN, ALL, ANY, EXISTS (3)

- Tìm các nhân viên có lương lớn hơn lương của các nhân viên phòng số 5.

```
select *
from NHANVIEN
where Luong > ALL (select Luong from NHANVIEN where MaPB=5);
```

- Tìm các nhân viên không tham gia các dự án do phòng số 5 điều phối.

```
select MaNV
from NHANVIEN
where MaNV not in (select TG.MaNV
                   from THAMGIA as TG, DUAN as DA
                   where TG.MaDA = DA.MaDA and
                         DA.MaPB = 5);
```

Truy vấn lồng phân cấp

- Mệnh đề WHERE của lệnh truy vấn con không tham chiếu đến các cột của các bảng trong mệnh đề FROM của lệnh truy vấn cha.
- Khi thực thi, lệnh truy vấn con được thực thi trước.
- Ví dụ

```
select *
from NHANVIEN
where Luong > ALL (select Luong from NHANVIEN
                   where MaPB = 5);
```

Truy vấn lồng tương quan (1)

- Mệnh đề WHERE của lệnh truy vấn con tham chiếu đến các cột của các bảng sau trong mệnh đề FROM của lệnh truy vấn cha.
- Khi thực hiện, lệnh truy vấn con được thực hiện nhiều lần, mỗi lần tương ứng với một dòng của lệnh truy vấn cha.
- Tìm các nhân viên không có thân nhân nào

```
select *
from NHANVIEN as NV
where not exists (select * from THANNHAN as TN
                  where TN.MaNV = NV.MaNV);
```

Truy vấn lồng tương quan (2)

NHANVIEN	MaNV	...	Luong
	888665555		55000
	333445555		40000
	123456789		30000

THANNHAN	MaNV	Ten	...
	333445555	Anh	
	333445555	The	
	987654321	An	
	123456789	Minh	
	123456789	Anh	

MaNV	Ten	...
333445555	Anh	
333445555	The	

MaNV	...	Luong
888665555		55000

IN và EXISTS

- IN
  - <tên cột> IN <lệnh truy vấn con>
  - Cột ở mệnh đề SELECT của lệnh truy vấn con phải cùng kiểu với <tên cột> ở mệnh đề WHERE của lệnh truy vấn cha
- EXISTS
  - Không cần có cột, hằng số hay biểu thức nào khác đứng trước.
  - Không nhất thiết liệt kê tên cột ở mệnh đề SELECT của lệnh truy vấn con.
  - Những lệnh truy vấn sử dụng so sánh =ANY hay IN đều có thể chuyển thành lệnh truy vấn sử dụng EXISTS.

Kết nối các bảng với JOIN

- JOIN hoặc INNER JOIN
  - Dùng kết nối hai bảng trong mệnh đề FROM.
  - SELECT <dsc>
    FROM (<bảng 1> JOIN <bảng 2> ON <đkk>)
- Cho biết họ tên, địa chỉ các nhân viên trong phòng Nghiên cứu

```
select HoNV, Dem, TenNV, DiaChi
from (NHANVIEN as NV join PHONGBAN as PB on
      NV.MaPB=PB.MaPB)
where TenPB='Nghien cuu';
```

- Các kết nối JOIN có thể lồng nhau
  - ((Q join R on <đkk>) join S on <đkk>)

Kế nối các bảng với OUTER JOIN (1)

- OUTER JOIN
  - Dùng để mở rộng cách kết nối hai bảng
    - LEFT JOIN hoặc LEFT OUTER JOIN.
    - RIGHT JOIN hoặc RIGHT OUTER JOIN.
    - FULL JOIN hoặc FULL OUTER JOIN.
- LEFT JOIN hay LEFT OUTER JOIN
  - Các dòng của bảng ở bên trái phải xuất hiện trong kết quả nếu không có dòng nào của bảng bên phải thỏa điều kiện.
  - Các cột của bảng bên phải trong kết quả sẽ nhận giá trị NULL.

Kế nối các bảng với OUTER JOIN (2)

- R LEFT JOIN S ON C<D

R	A	B	C
	α	α	1
	α	β	5
	β	β	12
	β	β	23

S	D	E
	1	7
	2	7
	12	3
	23	10

A	B	C	D	E
α	α	1	2	7
α	α	1	12	3
α	α	1	23	10
α	β	5	12	3
α	β	5	23	10
β	β	12	23	10
β	β	23	null	null

- Với mỗi nhân viên cho biết họ tên nhân viên và mã số của người giám sát nhân viên đó

```
select NV.HoNV, NV.Dem, NV.TenNV, GS.MaNV
from (NHANVIEN as NV LEFT JOIN NHANVIEN as GS on
      NV.MaGS=GS.MaNV);
```

Kế nối các bảng với OUTER JOIN (3)

- RIGHT JOIN hay RIGHT OUTER JOIN
  - Các dòng của bảng ở bên phải phải xuất hiện trong kết quả nếu không có dòng nào của bảng bên trái thỏa điều kiện.
  - Các cột của bảng bên trái trong kết quả sẽ nhận giá trị NULL.
- FULL JOIN hay FULL OUTER JOIN: tổ hợp kết quả của LEFT JOIN và RIGHT JOIN.

Tạo bảng tạm với WITH (1)

- WITH
  - Để tạo một bảng tạm chỉ được sử dụng trong một lệnh truy vấn cụ thể.
  - Bảng tạm sẽ bị hủy sau khi lệnh truy vấn được thực thi.
- Cú pháp

  WITH <định nghĩa bảng tạm> AS (<lệnh truy vấn>)

  <lệnh truy vấn tham chiếu đến bảng tạm>;

Tạo bảng tạm với WITH (2)

- Với mỗi phòng ban có nhiều hơn 3 nhân viên, hãy truy xuất mã số phòng ban và số lượng nhân viên có mức lương lớn hơn 30.000.

```
with PB5NV(MaPB) as (select MaPB
                     from NHANVIEN
                     group by MaPB
                     having count(*) > 3)
select MaPB, count(*)
from NHANVIEN
where Luong > 30000 and MaPB in (select * from PB5NV)
group by MaPB;
```

Cấu trúc CASE (1)

- Sử dụng khi một giá trị có thể khác dựa trên những điều kiện nhất định.
- CASE có thể sử dụng trong các lệnh SELECT, UPDATE, DELETE hoặc trong các mệnh đề WHERE, ORDER BY, HAVING.
- Cú pháp

```
CASE <tên cột>
    WHEN <giá trị> THEN <biểu thức>
    WHEN <giá trị> THEN <biểu thức>
    …
    [ELSE <biểu thức>]
END
```

Cấu trúc CASE (2)

- Cho biết họ tên, ngày sinh các nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
select HoNV, Dem, TenNV, NgaySinh
from NHANVIEN
where year(cast(getdate() as Date)) -
        year(NgaySinh) >= case GioiTinh
                            when 'M' then 60
                            when 'F' then 55
                          end;
```

Khung nhìn (View)

- Bảng (quan hệ) được tổ chức lưu trữ vật lý trong cơ sở dữ liệu.
- Khung nhìn là một bảng ảo được dẫn xuất từ các bảng, nhưng:
  - Không được lưu trữ vật lý trong cơ sở dữ liệu.
  - Không chứa dữ liệu.
  - Có thể truy vấn hay cập nhật dữ liệu thông qua khung nhìn.
- Sử dụng khung nhìn giúp
  - Che dấu tính phức tạp của dữ liệu.
  - Đơn giản hóa các câu truy vấn.
  - Hiển thị dữ liệu dưới dạng tiện dụng nhất.
  - An toàn dữ liệu.

Tạo và xóa khung nhìn (1)

- Tạo khung nhìn

  CREATE VIEW <tên khung nhìn>
  AS <lệnh truy vấn>
- Khung nhìn này có
  - Danh sách các cột là các cột sau mệnh đề SELECT của <lệnh truy vấn> tạo khung nhìn.
  - Các dòng chứa dữ liệu được lấy từ các bảng sau mệnh đề FROM của <lệnh truy vấn> tạo khung nhìn.
- Hủy khung nhìn

  DROP VIEW <tên khung nhìn>

Tạo và xóa khung nhìn (2)

- Tạo khung nhìn chứa các thông tin họ tên nhân viên và tên các dự án và số giờ mà nhân viên đó tham gia

```
create view NV_DA
as select NV.HoNV, NV.Dem, NV.TenNV,
          DA.TenDA, TG.SoGio
   from NHANVIEN as NV, DUAN as DA,
        THAMGIA as TG
   where NV.MaNV=TG.MaNV and
         TG.MaDA=DA.MaDA;
```

Tạo và xóa khung nhìn (3)

- Tạo khung nhìn chứa các thông tin mã số, tên của phòng và tổng số nhân viên, tổng mức lương của tất cả nhân viên của phòng ban đó.

```
create view PB_SoNV_Luong
as select PB.MaPB, PB.TenPB, count(*) as SoNV,
          sum(NV.Luong) as TongLuong
   from PHONGBAN as PB, NHANVIEN as NV
   where PB.MaPB=NV.MaPB
   group by PB.MaPB, PB.TenPB;
```

Truy vấn trên khung nhìn

- Cho biết họ tên, tên các dự án mà nhân viên đó tham gia với số giờ trên 20.

```
select HoNV, Dem, TenNV, TenDA
from NV_DA
where SoGio > 20;
```

- Cho biết tên dự án, tên phòng và tổng số nhân viên của phòng điều phối dự án đó.

```
select T1.TenDA, T2.TenPB, T2.SoNV
from DUAN as T1, PB_SoNV_Luong as T2
where T1.MaPB = T2.MaPB;
```

Cập nhật trên khung nhìn

- Có thể dùng các câu lệnh INSERT, DELETE, UPDATE cho các khung nhìn được dẫn xuất từ một bảng và chứa khóa chính của bảng.
- Không thể cập nhật dữ liệu nếu
  - Khung nhìn được dẫn xuất từ nhiều bảng.
  - Khung nhìn có dùng từ khóa DISTINCT.
  - Khung nhìn có sử dụng các hàm tập hợp.
  - Khung nhìn có mệnh đề SELECT mở rộng.
  - Khung nhìn được dẫn xuất từ bảng có ràng buộc trên cột.